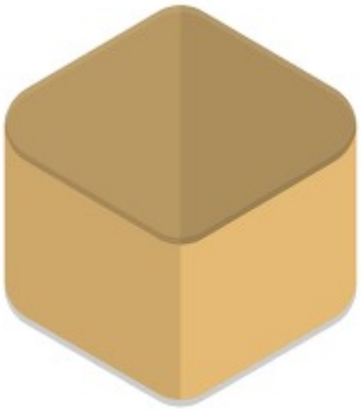
Cómo crear aplicaciones flatpak.

Traducido por Haylem Candelario Bauzá. 23/Julio/2023.
Tomado de: https://docs.flatpak.org/en/latest/getting-started.html



Empezando

Esta sección incluye una introducción a los conceptos básicos de Flatpak, una guía sobre cómo usar la interfaz de línea de comandos de Flatpak y un tutorial para crear una aplicación sencilla. Para completar esta sección, Flatpak debería haberse instalado y el repositorio de Flathub debería estar habilitado. El sitio web de Flatpak proporciona instrucciones sobre cómo hacer esto con una variedad de distribuciones.

Conceptos básicosConceptos básicos
- ***Tiempos de ejecución***
- ***Bibliotecas agrupadas***
- ***cajas de arena***
- ***Portales***
- ***Repositorios***

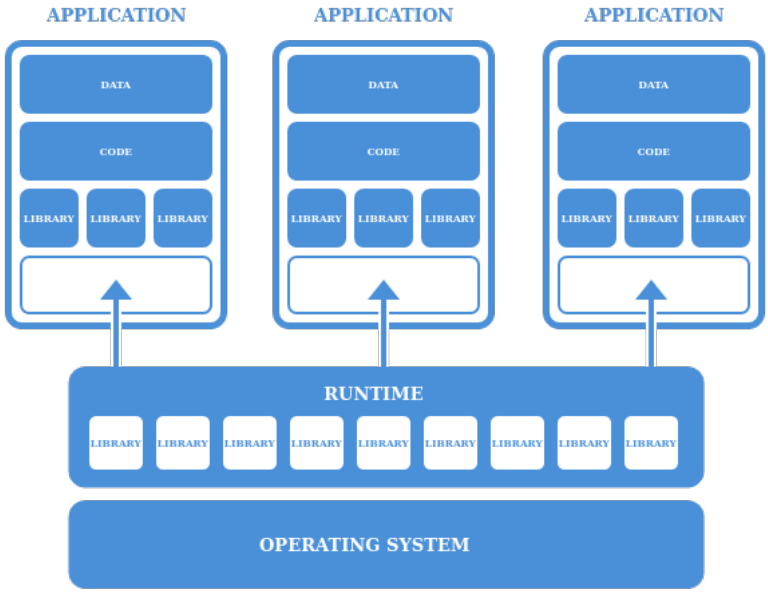
Usando Flatpak
- ***El comando flatpack***
- ***Identificadores***
- ***Sistema versus usuario***
- ***Comandos básicos***

Construyendo tu primer Flatpak
- ***1. Instale un tiempo de ejecución y el SDK correspondiente***
- ***2. Crea la aplicación***
- ***3. Agregar un manifiesto***
- ***4. Cree la aplicación***
- ***5. Prueba la construcción***
- ***6. Pon la aplicación en un repositorio***
- ***7. Instala la aplicación***

8. Ejecute la aplicación

Conceptos básicos

Flatpak se puede entender a través de una pequeña cantidad de conceptos clave. Es útil familiarizarse con estos antes de aprender a usar Flatpak desde la línea de comandos o usarlo para crear aplicaciones.



Tiempos de ejecución

Los tiempos de ejecución proporcionan las dependencias básicas que utilizan las aplicaciones. Cada aplicación debe construirse contra un tiempo de ejecución, y este tiempo de ejecución debe instalarse en un sistema host para que la aplicación se ejecute (Flatpak puede instalar automáticamente el tiempo de ejecución requerido por una aplicación). Se pueden instalar varios tiempos de ejecución diferentes al mismo tiempo, incluidas diferentes versiones del mismo tiempo de ejecución.

Los tiempos de ejecución son independientes de la distribución y no dependen de versiones de distribución particulares. Esto significa que proporcionan una base de distribución cruzada estable para las aplicaciones y permiten que las aplicaciones continúen funcionando independientemente de las actualizaciones del sistema operativo.
Bibliotecas agrupadas

Si una aplicación requiere dependencias que no están en su tiempo de ejecución, se pueden agrupar como parte de la aplicación. Esto brinda a los desarrolladores de aplicaciones flexibilidad con respecto a las dependencias que usan, incluido el uso de:

- B***ibliotecas que no están disponibles en un tiempo de ejecución***
- ***Diferentes versiones de bibliotecas de las que están en tiempo de ejecución***
- ***Versiones parcheadas de bibliotecas***

Cajas de arena

Con Flatpak, cada aplicación se crea y ejecuta en un entorno aislado, que se denomina "sandbox". Cada sandbox contiene una aplicación y su tiempo de ejecución. De forma predeterminada, la aplicación solo puede acceder al contenido de su sandbox. El acceso a los archivos de usuario, la red, los sockets de gráficos, los subsistemas en el bus y los dispositivos debe otorgarse explícitamente. El acceso a otras cosas, como otros procesos, no es posible deliberadamente.

Por necesidad, algunos recursos que están dentro de la zona de pruebas deben exponerse al exterior para que los utilice el sistema host. Estos se conocen como "exportaciones", ya que son archivos que se exportan fuera de la zona de pruebas e incluyen elementos como el archivo y el icono .desktop de la aplicación.

Portales

Los portales son un mecanismo a través del cual las aplicaciones pueden interactuar con el entorno del host desde dentro de un espacio aislado. Brindan la capacidad de interactuar con datos, archivos y servicios sin la necesidad de agregar permisos de sandbox.

Los ejemplos de capacidades a las que se puede acceder a través de los portales incluyen la apertura de archivos a través de un cuadro de diálogo de selección de archivos o la impresión. Los kits de herramientas de interfaz pueden implementar un soporte transparente para los portales, por lo que el acceso a los recursos fuera de la zona de pruebas funcionará de forma segura y lista para usar.

Puede encontrar más información sobre los portales en Permisos de Sandbox.

Repositorios

Las aplicaciones y los tiempos de ejecución de Flatpak generalmente se almacenan y publican mediante repositorios, que se comportan de manera muy similar a los repositorios de Git. Un repositorio de Flatpak puede contener un solo objeto o varios objetos, y cada objeto está versionado, lo que permite actualizar e incluso degradar.

Cada sistema que utiliza Flatpak se puede configurar para acceder a cualquier cantidad de repositorios remotos. Una vez que se ha configurado un sistema para acceder a un "remoto", el contenido del repositorio remoto se puede inspeccionar y buscar, y se puede utilizar como fuente de aplicaciones y tiempos de ejecución.

Cuando se realiza una actualización, las nuevas versiones de las aplicaciones instaladas y los tiempos de ejecución se descargan desde los controles remotos correspondientes. Al igual que con Git, solo se descarga la diferencia entre versiones, lo que hace que el proceso sea muy eficiente.

Usando Flatpak

Esta página proporciona una introducción a la interfaz de línea de comandos de flatpak y explica las convenciones técnicas esenciales, así como los comandos más comunes.

Los usuarios finales generalmente no deberían necesitar usar esta página o la interfaz de línea de comandos de Flatpak, ya que Flatpak se puede usar fácilmente a través de herramientas gráficas de administración de software, ¡aunque, por supuesto, pueden usar la línea de comandos si lo prefieren!

El comando flatpack

flatpak es el comando principal de Flatpak, al que se agregan comandos específicos. Por ejemplo, el comando para instalar algo es ***`flatpak install`*** `y el comando para desinstalar es` ***`flatpak uninstall`***`.`

`Identificadores`

Flatpak identifica cada aplicación y tiempo de ejecución mediante un identificador único de tres partes, como com.company.App. El segmento final de esta dirección es el nombre del objeto y la parte anterior identifica al desarrollador, de modo que el mismo desarrollador puede tener varias aplicaciones, como com.company.App1 y com.company.App2.

El identificador se triplica

Por lo general, es suficiente referirse a los objetos usando su ID. Sin embargo, en algunas situaciones es necesario referirse a una versión específica de un objeto o a una arquitectura específica. Por ejemplo, algunas aplicaciones pueden estar disponibles como versión estable y de prueba, en cuyo caso es necesario especificar cuál desea instalar.

Flatpak permite especificar arquitecturas y versiones utilizando el identificador triple de un objeto. Esto toma la forma de nombre/arquitectura/sucursal, como com.company.App/i386/stable. (Rama es el término que se usa para referirse a las versiones del mismo objeto). La primera parte del triple es el ID, la segunda parte es la arquitectura y la tercera parte es la rama.

Los triples de identificadores también se pueden usar para especificar solo la arquitectura o la rama, dejando parte del triple en blanco. Por ejemplo, com.company.App//stable solo especificaría la sucursal y com.company.App/i386// solo especificaría la arquitectura.

La CLI de Flatpak proporciona información si es necesario especificar la arquitectura o la rama de un objeto.

Sistema versus usuario

Los comandos de Flatpak se pueden ejecutar en todo el sistema o por usuario. Las aplicaciones y los tiempos de ejecución que se instalan en todo el sistema están disponibles para todos los usuarios del sistema. Las aplicaciones y los tiempos de ejecución que se instalan por usuario solo están disponibles para los usuarios que los instalaron.

El mismo principio se aplica a los repositorios: los repositorios que se han agregado en todo el sistema están disponibles para todos los usuarios, mientras que los repositorios por usuario solo pueden ser utilizados por un usuario en particular.

Los comandos de Flatpak se ejecutan en todo el sistema de forma predeterminada. Si está instalando aplicaciones para el uso diario, se recomienda mantener este comportamiento predeterminado.

Sin embargo, ejecutar comandos por usuario puede ser útil para fines de prueba y desarrollo, ya que los objetos que se instalan de esta manera no estarán disponibles para otros usuarios en el sistema. Para hacer esto, use la opción --user, que se puede usar en combinación con la mayoría de los comandos de flatpak.

Los comandos se comportan exactamente de la misma manera si se ejecutan por usuario en lugar de en todo el sistema.

Comandos básicos

Esta sección cubre los comandos básicos necesarios para instalar, ejecutar y administrar aplicaciones Flatpak. Para ver la lista completa de comandos de Flatpak, ejecute flatpak --help o consulte la Referencia de comandos de Flatpak.

Lista de controles remotos

Para enumerar los controles remotos que ha configurado en su sistema, ejecute:

```
$ flatpak remotes
```

Esto da una lista de los controles remotos existentes que se han agregado. La lista indica si cada control remoto se agregó por usuario o en todo el sistema.

Agregar un control remoto

La forma más conveniente de agregar un control remoto es usando un archivo .flatpakrepo, que incluye tanto los detalles del control remoto como su clave GPG:

$ flatpak remote-add --if-not-exists flathub https://flathub.org/repo/flathub.flatpakrepo

Aquí, flathub es el nombre local que se le da al control remoto. La URL apunta al archivo .flatpakrepo del control remoto. --if-not-exists evita que el comando produzca un error si el control remoto ya existe.

Quitar un control remoto

Para eliminar un control remoto, ejecute:

```
$ flatpak remote-delete flathub
```

En este caso, flathub es el nombre local del control remoto.

Buscar

Applications can be found in any of your remotes using the search command. For example:

$ flatpak search gimp

La búsqueda devolverá cualquier aplicación que coincida con los términos de búsqueda. Cada resultado de búsqueda incluye el ID de la aplicación y el control remoto en el que se encuentra la aplicación. En este ejemplo, el término de búsqueda es gimp.

Instalar aplicaciones

Para instalar una aplicación, ejecute:

$ flatpak install flathub org.gimp.GIMP

Aquí, flathub es el nombre del control remoto desde el que se instalará la aplicación, y org.gimp.GIMP es el ID de la aplicación.

A veces, una aplicación requerirá un tiempo de ejecución particular, y este se instalará antes que la aplicación.

Los detalles de la aplicación que se instalará también se pueden proporcionar mediante un archivo .flatpakref, que puede ser remoto o local. Para especificar un .flatpakref en lugar de proporcionar manualmente el ID remoto y de la aplicación, ejecute:

$ flatpak install https://flathub.org/repo/appstream/org.gimp.GIMP.flatpakref

Si el archivo .flatpakref especifica que la aplicación se instalará desde un control remoto que aún no se haya agregado, se le preguntará si desea agregarlo antes de instalar la aplicación.

Desde Flatpak 1.2, el comando de instalación puede buscar aplicaciones. Un simple:

$ flatpak install gimp

confirmará el control remoto y la aplicación y procederá a la instalación.

Ejecutando aplicaciones

Una vez que se ha instalado una aplicación, se puede iniciar con el comando de ejecución y su ID de aplicación:

$ flatpak run org.gimp.GIMP

Actualizando

Para actualizar todas sus aplicaciones instaladas y tiempos de ejecución a la última versión, ejecute:

$ flatpak update

Lista de aplicaciones instaladas

Para enumerar las aplicaciones y tiempos de ejecución que ha instalado, ejecute:

$ flatpak list

Alternativamente, para enumerar las aplicaciones instaladas, ejecute:

$ flatpak list –app

Quitar una aplicación

Para eliminar una aplicación, ejecute:

$ flatpak uninstall org.gimp.GIMP

Solución de problemas

Flatpak tiene algunos comandos que pueden ayudarlo a que todo vuelva a funcionar cuando algo sale mal.

Para eliminar tiempos de ejecución y extensiones que no utilizan las aplicaciones instaladas, utilice:

$ flatpak uninstall –unused

Para corregir inconsistencias con su instalación local, use:

$ flatpak repair

Flatpak también tiene varios comandos para administrar los permisos del portal de las aplicaciones instaladas. Para restablecer todos los permisos del portal para una aplicación, use *flatpak permission-reset:*

$ flatpak permission-reset org.gimp.GIMP

Para averiguar qué cambios se han realizado en su instalación de Flatpak a lo largo del tiempo, puede consultar los registros (desde 1.2):

$ flatpak history

Construyendo tu primer Flatpak

Este tutorial proporciona una introducción rápida a la creación de Flatpaks. En él, aprenderá a crear una aplicación Flatpak básica, que puede instalarse y ejecutarse.

Para completar este tutorial, debería haber seguido la guía de configuración en flatpak.org. También es necesario tener instalado ***flatpak-builder*** que generalmente está disponible en el mismo repositorio que el paquete flatpak (por ejemplo, use ***apt*** o ***dnf***). También puedes instalarlo como flatpak con:

flatpak install flathub org.flatpak.Builder

1. Instale un tiempo de ejecución y el SDK correspondiente

Flatpak requiere que cada aplicación especifique un tiempo de ejecución que utiliza para sus dependencias básicas. Cada tiempo de ejecución tiene un SDK (Kit de desarrollo de software) correspondiente, que contiene todo lo que hay en el tiempo de ejecución, además de encabezados y herramientas de desarrollo. Este SDK es necesario para crear aplicaciones para el tiempo de ejecución.

En este tutorial usaremos el tiempo de ejecución y el SDK de Freedesktop 22.08. Para instalarlos, ejecute:

$ flatpak install flathub org.freedesktop.Platform//22.08 org.freedesktop.Sdk//22.08

2. Crea la aplicación

La aplicación que se creará para este tutorial es un script simple. Para crearlo, copie lo siguiente:

```
#!/bin/sh
echo "Hello world, from a sandbox"
```

Ahora pegue esto en un archivo vacío y guárdelo como hello.sh.

3. Agregar un manifiesto

Cada Flatpak se construye utilizando un archivo de manifiesto que proporciona información básica sobre la aplicación e instrucciones sobre cómo se debe construir. Para agregar un manifiesto a la aplicación hello world, agregue lo siguiente a un archivo vacío:

```
app-id: org.flatpak.Hello
runtime: org.freedesktop.Platform
runtime-version: '22.08'
sdk: org.freedesktop.Sdk
command: hello.sh
modules:
  - name: hello
    buildsystem: simple
    build-commands:
      - install -D hello.sh /app/bin/hello.sh
    sources:
      - type: file
        path: hello.sh
```

Ahora guarde el archivo junto con hello.sh y llámelo org.flatpak.Hello.yml.

En una aplicación más compleja, el manifiesto enumeraría varios módulos. Por lo general, el último sería la aplicación en sí, y los anteriores serían dependencias que se incluyen con la aplicación porque no forman parte del tiempo de ejecución.

4. Cree la aplicación

Ahora que la aplicación tiene un manifiesto, se puede usar flatpak-builder para compilarla. Esto se hace especificando el archivo de manifiesto y un directorio de destino:

```
$ flatpak-builder build-dir org.flatpak.Hello.yml
```

Este comando compilará cada módulo que se enumera en el manifiesto y lo instalará en el subdirectorio /app, dentro del directorio build-dir.

5. Prueba la construcción

Para verificar que la compilación fue exitosa, ejecute lo siguiente:

$ flatpak-builder --user --install --force-clean build-dir org.flatpak.Hello.yml
$ flatpak run org.flatpak.Hello

Esta segunda vez pasamos --force-clean, lo que significa que el directorio build-dir creado anteriormente se eliminó antes de que se iniciara la nueva compilación.

¡Felicitaciones, has creado una aplicación!

6. Pon la aplicación en un repositorio

Si desea compartir la aplicación, puede ponerla en un repositorio. Esto se hace pasando el argumento --repo a flatpak-builder:

$ flatpak-builder --repo=repo --force-clean build-dir org.flatpak.Hello.yml

Esto hace la compilación nuevamente y al final exporta el resultado a un directorio local llamado repo. Tenga en cuenta que **flatpak-builder** guarda un caché de compilaciones anteriores en el subdirectorio .flatpak-builder, por lo que hacer una segunda compilación como esta es muy rápido.

Para que su aplicación aparezca en las tiendas de aplicaciones mientras se prueba con un repositorio local, es posible que deba ejecutar *flatpak build-update-repo repo* . Para obtener más información sobre cómo publicar en las tiendas de aplicaciones, consulte Archivos de datos de aplicaciones.

7. Instala la aplicación

Ahora estamos listos para agregar el repositorio que acabamos de crear e instalar la aplicación. Esto se hace con dos comandos:

$ flatpak --user remote-add --no-gpg-verify tutorial-repo repo
$ flatpak --user install tutorial-repo org.flatpak.Hello

El primer comando agrega el repositorio que se creó en el paso anterior. El segundo comando instala la aplicación desde el repositorio.

Ambos comandos usan el argumento --user, lo que significa que el repositorio y la aplicación se agregan por usuario en lugar de en todo el sistema. Esto es útil para las pruebas.

Tenga en cuenta que el repositorio se agregó con --no-gpg-verify, ya que no se especificó una clave GPG cuando se creó la aplicación. Esto está bien para realizar pruebas, pero para los repositorios oficiales debe firmarlos con una clave GPG privada.

8. Ejecute la aplicación

Todo lo que queda es probar la aplicación. Esto se puede hacer con el siguiente comando:

$ flatpak run org.flatpak.Hello

Esto ejecuta la aplicación, de modo que imprime 'Hola mundo, desde una caja de arena'.